WWW.SCIAM-MAGAZINE.NAROD.RU

том, что для возбуждения краевых волн требуются довольно большие затраты энергии. Если я продолжал уменьшать напряжение, то краевые волны также начинали медленно уменьшаться. Время, за которое они исчезали, было примерно таким же, как и время, за которое они устанавливались.

Я проследил за периодическим обменом энергией между краевыми волнами и пластинкой в установившемся режиме. Для того чтобы видеть реальную картину гребней, я включил комнатное освещение, а затем измерил время между двумя моментами — одним, когда высота гребня была наибольшей, и другим, когда эта высота была наименьшей. Это время оказалось равным 14 с.

После этого я налил на поверхность воды немного кукурузного масла. Первые несколько капель растеклись и образовали тонкий слой. Потом масло собралось в виде больших пятен, толщина которых постепенно достигла нескольких миллиметров.

В течение нескольких минут после того, как я налил масло, краевые волны оставались прежними. Затем они исчезли. Сначала я решил, что вязкость масла слишком велика. Потом мне пришло в голову, что в исчезновении волн виновата смесь капелек воды и пузырьков воздуха в масле, взбитая пластинкой. Проверить догадку я смог, подождав, пока смесь не рассосалась и слой масла не стал однородным. Когда я после этого включил мотор, краевые волны образовались вновь.

Могут ли краевые волны возникать на границе между водой и маслом? Частично выдвинув блюдо за край стола, я получил возможность наблюдать через стеклянное дно за границей раздела. Краевые волны не появлялись там даже в том случае, когда на поверхности масла они были весьма заметны.

Наконец я вылил на поверхность воды целый флакон одной лечебной мази. На сей раз краевые волны не появились. То ли мазь была слишком вязкой, то ли мотор слишком слабым для того, чтобы пластинка могла сильно колебаться в такой густой среде.

Понаблюдайте за краевыми волнами в других жидкостях. Возможно, вам удастся обнаружить их на границе раздела между двумя несмешивающимися жидкостями. Может оказаться, однако, что такая граница раздела не в состоянии поддерживать волновое движение.

ДИФРАКЦИОННЫЕ картинки, которые я предложил читателям расшифровать в предыдущем номере журнала, образованы буквами M O R I S N. Я имел в виду Филипа Моррисона, ведущего рубрику «Книги» в журнале «Scientific American».

Занимательный компьютер

Акулы и рыбы ведут экологическую войну на тороидальной планете Аква-Тор

А. К. ДЬЮДНИ

ГДЕ-ТО в «стране развлечений», далеко-далеко — так далеко, как только может вообразить себе самый искусный программист, среди звезд проплывает планета Аква-Тор. Она имеет форму тора, или бублика, полностью покрытого водой. Основные обитатели Аква-Тора — акулы и рыбы, названные так по наиболее похожим на них земным животным. Акулы поедают рыб, а для рыб на Аква-Торе всегда вдоволь пищи.

Такая простейшая экосистема могла бы показаться стабильной, почти застывшей, если бы не тот факт, что численность акул и рыб испытывает сильнейшие колебания. В прошлом неоднократно случалось так, что почти все рыбы были съедены, тогда как в другие периоды акулы почти вымирали от голода (даже при наличии большого количества рыб). Тем не менее и акулы, и рыбы выжили. Чтобы понять, почему так происходит, я разработал программу для моделирования процессов питания и размножения в этой системе.

Однако, прежде чем мне довелось увидеть эти экологические ритмы на экране дисплея (см. рисунок на с. 80), я провел немало времени в размышлениях об общих принципах и технических деталях программы АКВАТОР. Однажды за обедом ко мне за стол подсел Дэвид Визман, штатный «кудесник» по части системных программ в моем отделении Университета пров. Западный Онтарио. Рассказ о моем проекте вызвал загадочную улыбку на лице Магги (так называли Визмана). На следующее утро он гордо ввел меня в свой кабинет, чтобы продемонстрировать работающую программу.

«Смотри», — сказал он и нажал на клавишу. Сначала рыбы и акулы медленно блуждали от точки к точке вроде бы без всякой системы. Одни акулы, не сумев добыть пищу, исчезали, у других появлялось потомство, столь же прожорливое, как и родители. Несколько рыб, которым посчастливилось попасть в район без акул, расплодились и образовали огромный косяк. Вскоре некоторые акулы обнаружили этот косяк, расположились по его краям и набросились на рыбу, двигаясь внутрь косяка. Через несколько минут на экране появились статистические данные, показавшие, что в этот момент в системе было 578 рыб и 68 акул.

Какие-то люди заходили в кабинет Магги. Не прошло и пяти минут, как комната заполнилась. Все «болели» за акул. Постепенно кольцо из акул замкнулось вокруг несчастных рыб. В другой части экрана медленно разрастался, оставаясь незамеченным, еще один косяк рыб — пока небольшой. Когда большой косяк был окончательно уничтожен и акулы, умирая одна за другой, беспорядочно закружили в поисках добычи, по кабинету пронесся тяжелый вздох. Мне пришло в голову изменить правила так, чтобы акулы могли есть друг друга, но я понял, что это не продлило бы заметно их существование, и само появление второго косяка рыб было бы под вопросом. Наконец, две акулы наткнулись на этот косяк, и все началось сначала.

Программа для моделирования Аква-Тора не длинна и ее нетрудно составить. Тем из читателей, у кого есть персональный компьютер, стоит заняться реализацией этого проекта. Их усилия будут вознаграждены, когда программа будет написана, отлажена и запущена. Перед пуском можно задать такие параметры, как время до появления потомства, допустимое время голодания и начальный размер популяции. После этого остается только ждать и наблюдать, как первоначально неорганизованная «смесь» рыб и акул постепенно приобретает черты экосистемы.

В программу АКВАТОР заложено несколько правил поведения акул и рыб. Оба представителя фауны плавают по «океану», имеющему вид прямоугольной решетки, причем узлы на противоположных сторонах прямоугольника попарно отождествлены. Это означает, что если рыба или акула находится в одном из самых правых узлов решетки и решает поплыть на «восток» (вправо), то она вновь появится в соответствующем узле на левом краю. Та же связь имеется между верхней и нижней границами. Получающееся в результате свернутое двумерное пространство и есть, в сущности, тор, т.е. совпадает с действительной формой поверхности Аква-Тора (см. рисунок на с. 81). Размер решетки может выбираться по усмотрению составителей программы АКВАТОР. Например, в программе, разработанной Магги, которая выполнялась на

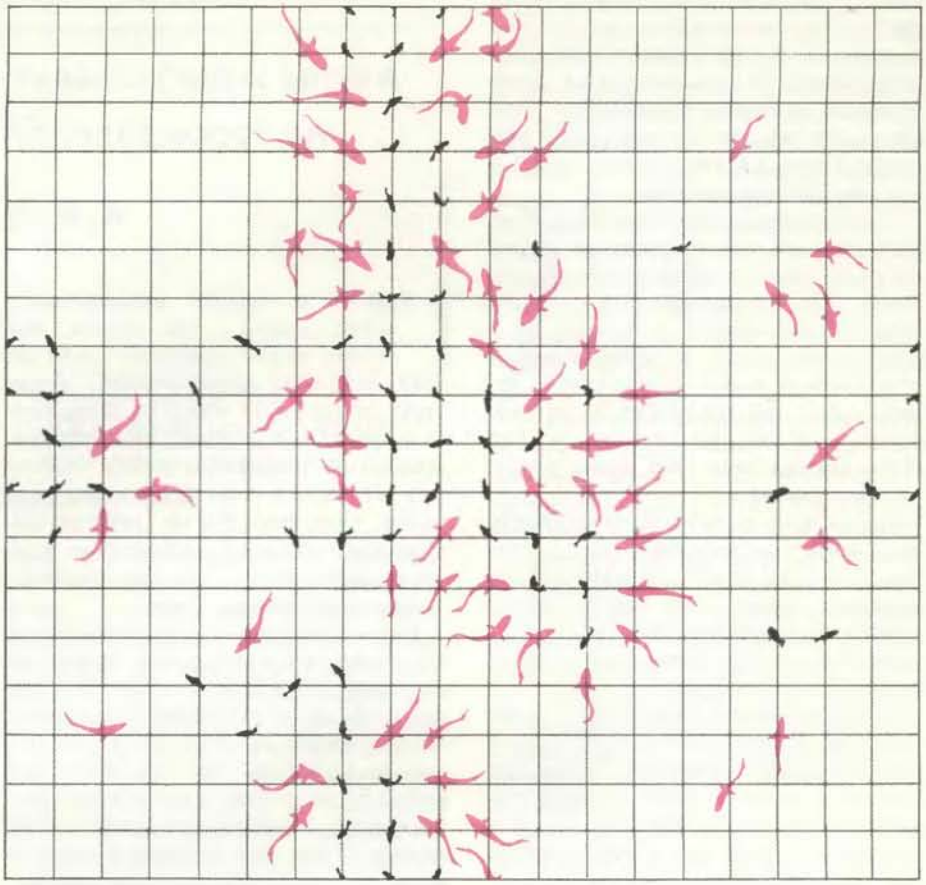
WWW.SCIAM-MAGAZINE.NAROD.RU

ЭВМ VAX, океан составлял 80 точек в ширину и 23 в высоту. В моей версии АКВАТОРа, написанной для персонального компьютера фирмы IBM, использовался более «скромный» по размерам океан — 32 на 14 точек.

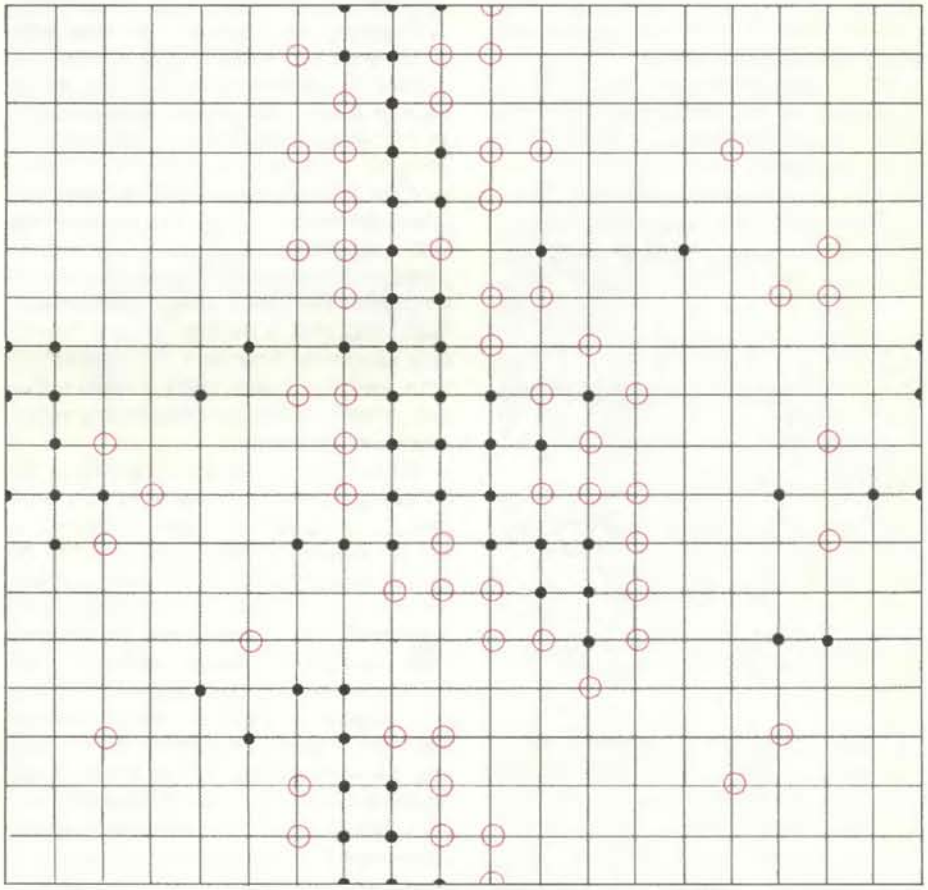
Согласно этой программе, время течет дискретными порциями, которые я назвал хрононами. За один хронон каждая рыба или акула может переместиться на «север», «восток», «юг» или «запад» в соседний узел решетки при условии, что этот узел не занят животным того же вида. Выбор конкретного перемещения осуществляется с помощью генератора случайных чисел. Для рыб правило простое — двигаться следует в один из соседних незанятых узлов, который выбирается случайным образом. Если все четыре соседних узла заняты, то рыба не двигается. Действия акул несколько сложнее, поскольку для них охота за рыбой предпочтительнее, чем простое перемещение. Акула выбирает случайно один из соседних узлов, в котором есть рыба, плывет туда и съедает ее. Если в ближайшей окрестности нет рыбы, то акула движется точно так же, как рыба, избегая своих собратьев-акул.

Для указания конкретной популяции составитель АКВАТОРа задает пять параметров. Параметры *nfish* и *nsharks* представляют собой начальное число рыб и акул. Программа случайным образом размещает указанное число рыб и акул более или менее равномерно по поверхности планеты. Параметры *fbreed* и *sbreed* показывают, сколько хрононов должна прожить соответственно рыба или акула до появления у нее одного потомка (оба вида размножаются, по-видимому, бесполым способом). Последний параметр, *starve*, определяет число хрононов, «отпущенное» акуле на поиски пищи. Если акула проплавает, не поев, хотя бы чуть большее время, она гибнет. На протяжении каждого хронона программа перемещает каждую рыбу и акулу и отображает полученный результат на экране дисплея. Реализовав эти простые правила, можно наблюдать все перипетии богатой кризисами экосистемы Аква-Тора.

Мы с Магги не раз были свидетелями такого хода событий (определяемого пятью перечисленными параметрами), когда океан Аква-Тора оказывался перенаселен рыбой, но заканчивалось все лишь тем, что размножающиеся акулы поедали всю рыбу и гибли сами. В других случаях мы наблюдали, как акулы съедали всю рыбу в одном большом косяке. Объевшиеся акулы в конце концов гибли, не заметив поблизости небольшой группы рыб. Иногда «хищники» и «жертвы» демонстрировали отношения взаимовыручки на протяжении двух или даже трех циклов, пока акулы не терпели окончательный крах. Одна-



Реалистическое изображение акул, поедающих рыб



Это изображение легче запрограммировать; кружочки соответствуют акулам, а точки — рыбам

WWW.SCIAM-MAGAZINE.NAROD.RU

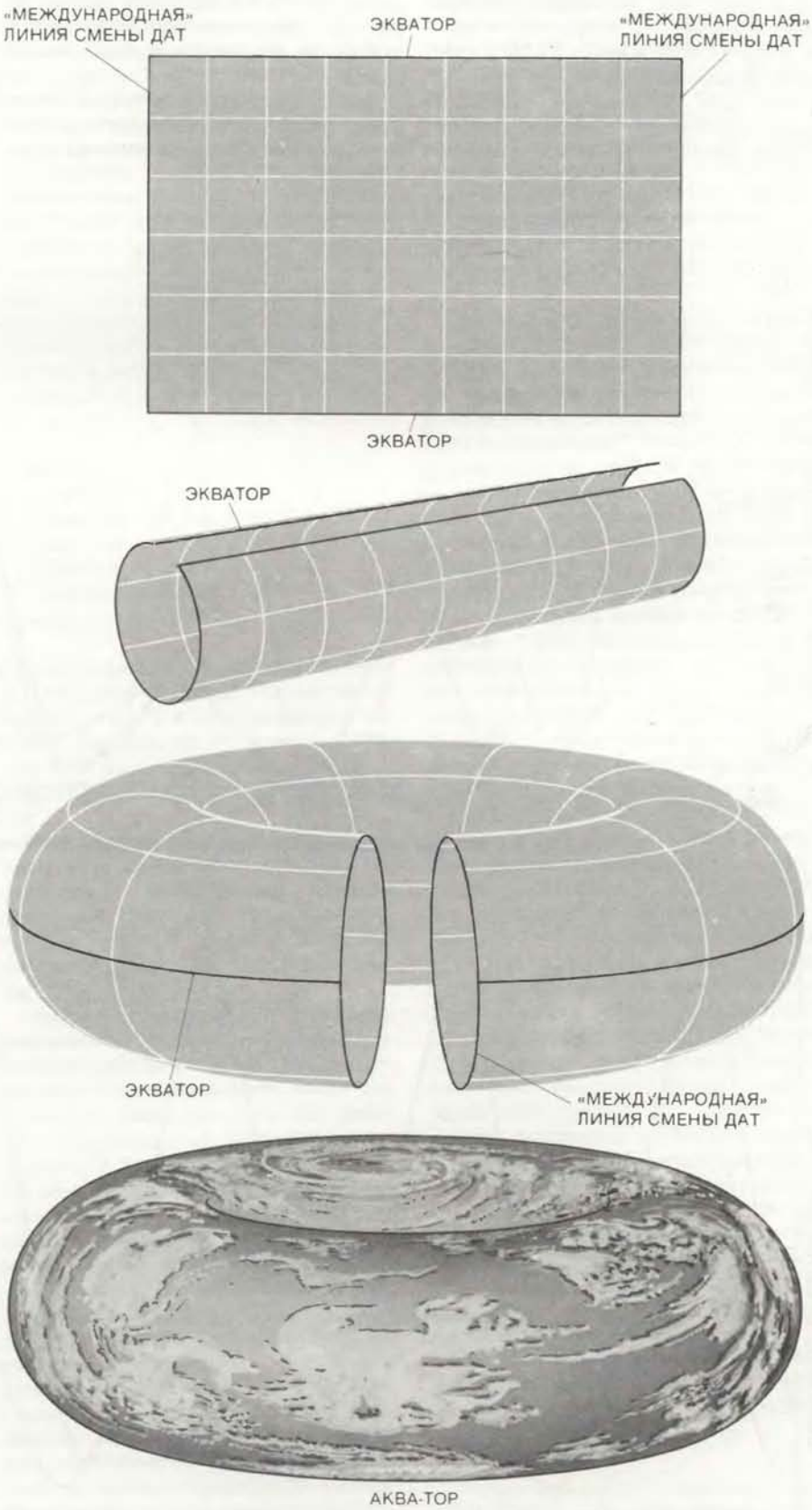
ко значения параметров, которые были выбраны в этих ситуациях, не давали ни малейшего намека на такие характеристики, которые обеспечивали бы «нетупиковую» экосистему. Как же удалось уцелеть обитателям Аква-Тора?

Говорят, что биология — это «судьба». Мы с Магги не можем устоять от искушения заявить, что экология — это геометрия, по крайней мере в случае планеты Аква-Тора. Окончательная участь какого-либо экологического сценария, как нам кажется, не зависит от начального случайного распределения акул и рыб. Не зависит она и от конкретного способа выполнения случайных перемещений. Однако вероятность гибели популяции оказывается тесно связанной с геометрией расположения акул и рыб, отчетливо проявляющейся на экране дисплея: чем более организованными и локализованными становятся популяции, тем более вероятно, что экосистема планеты обречена на вырождение. Размышляя на эту тему, мы задались вопросом: как следует выбрать пять параметров, чтобы разрушить геометрический порядок? Внезапно нас осенило: если уж акулы сконцентрировались по краям косяка рыбы, то нарушить эту геометрию можно лишь одним способом — заставить акул размножаться реже, поскольку само скопление образуется в большей степени благодаря размножению, а не перемещению акул.

Прежде чем прийти к такому выводу, мы задавали для акул и рыб примерно равные времена размножения. Мы полагали, что сбалансированные темпы размножения приведут к сбалансированным численностям видов. Не подобные ли рассуждения являются первопричиной многих бед нашего сегодняшнего технологического мира? Как бы то ни было, я поместил в свой океан размером 32 × 14 единиц 200 рыб и 20 акул и установил, что рыбы будут размножаться каждые три хронона, в то время как акулам придется подождать до истечения 10 хрононов. Время голодной смерти акул было более или менее произвольно выбрано равным трем хрононам. Пронаблюдав работу моей относительно медленной программы в течение 15 минут, мы увидели, что наши усилия не пропали даром: численность популяций полностью восстановилась после некоторого спада вначале. Более того, геометрический порядок, хотя и не отсутствовал совсем, был скорее предположительным, чем вполне определенным. Косяки представляли собой бесформенные скопления с неровными краями, а в некоторых местах экрана акулы и рыбы были перемешаны почти случайно.

Я запустил программу на все послеполуденное время и занялся более важными делами, отрываясь время от времени, чтобы взглянуть на нее. Программа проработала всю ночь, и когда я зашел в свое учреждение после утренней лекции, то обнаружил, что акулы и рыбы все еще продолжают свою циклическую жизнь. Это был настоящий Аква-Тор!

Программу АКВАТОР можно реализовать многими способами, но, пожалуй, самый простой из них основан на работе с рядом двумерных массивов. Я



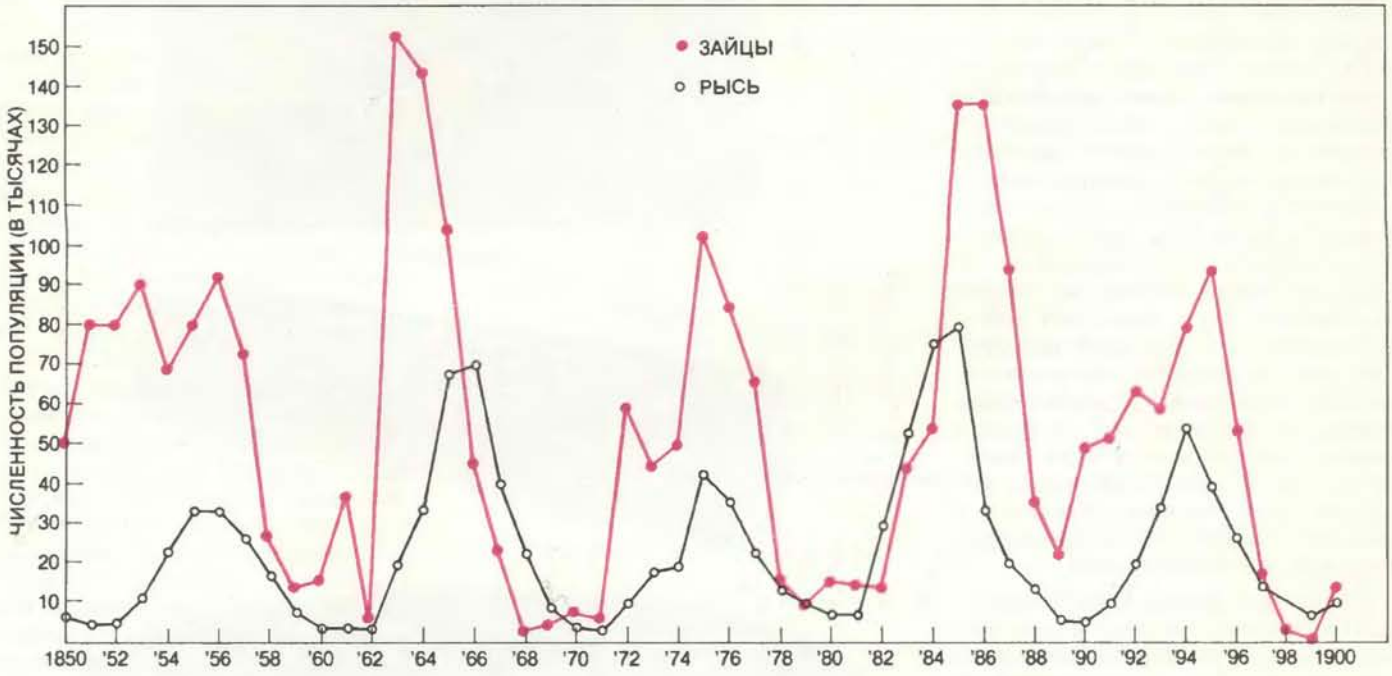
Тороидальная планета Аква-Тор и ее изображение на плоской карте (или на плоском экране дисплея)

WWW.SCIAM-MAGAZINE.NAROD.RU

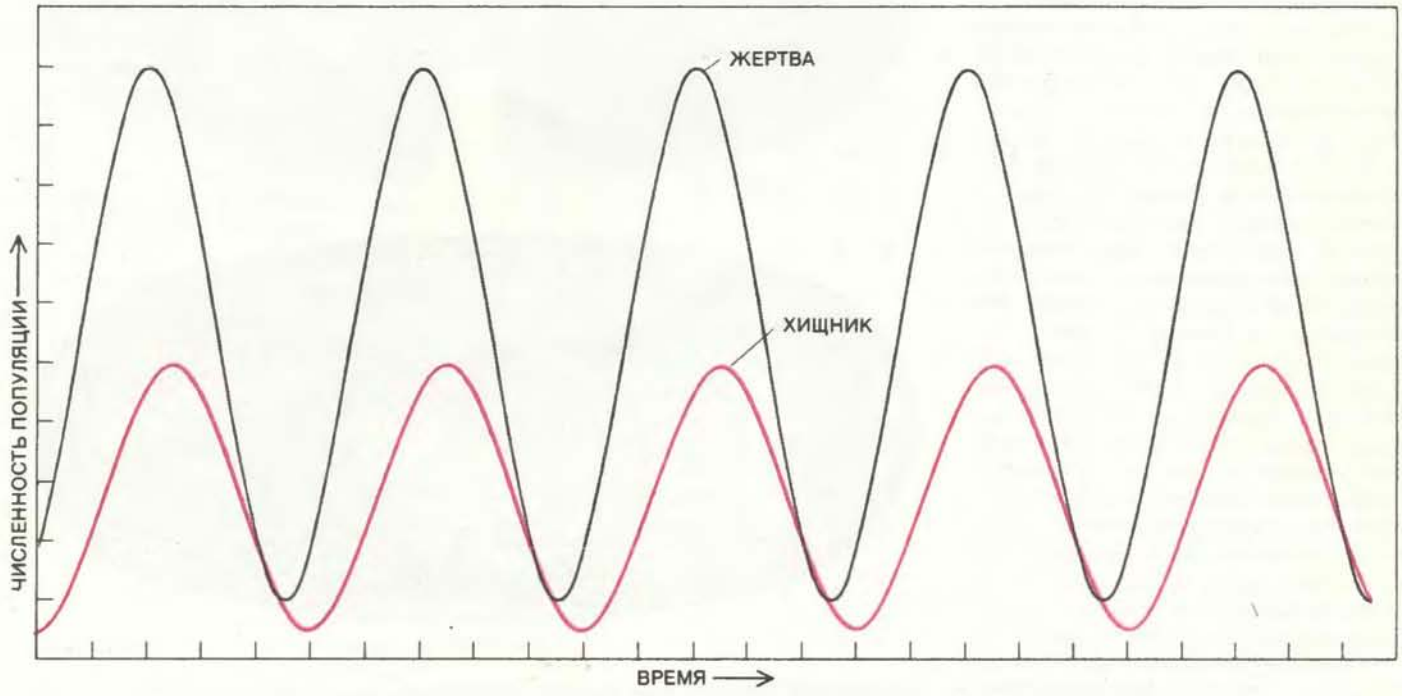
использую пять массивов — FISH, SHARKS, FISHMOVE, SHARKMOVE и STARVE. Эти массивы размером 32 на 14 хранят информацию о положении и возрасте рыб и акул. Более точно FISH (I, J) указывает наличие или отсутствие рыбы в узле с координатами (I, J). Если в узле рыбы нет, то элемент массива имеет значение −1. В противном случае в нем записан возраст рыбы в хрононах. В массиве SHARKS точно в таком же виде хранится информация о положении и возрасте акул. В каждой позиции массива FISHMOVE записано, переместилась ли туда рыба в текущем цикле вычислений. Эта информация позволяет программе избежать повторного перемещения рыбы в течение одного хронона. Массив SHARKMOVE выполняет ту же функцию по отношению к акулам. Массив STARVE хранит время, когда каждая акула последний раз ела. Если в некоторой позиции нет акулы, то соответствующий элемент равен −1.

Чтобы показать события на Аква-Торе, проще всего изобразить каждую строку массива строкой литер на экране дисплея: пробел в некоторой позиции означает, что она свободна, точка (.) изображает рыбу, а нуль (0) — акулу. Несмотря на кажущуюся ограниченность такого способа, он очень информативен и удобен для наблюдения.

Начальная фаза программы АКВАТОР состоит в равномерном «рассеивании» по тороидальному океану требуемого количества рыб и акул. Затем программа циклически выполняет три описанные ниже части, или подпрограммы; каждый такой цикл занимает как раз один хронон.



Число рысей и зайцев (в тысячах), добытых для Hudson's Bay Company с 1850 по 1900 г.



Теоретическое соотношение хищников и жертв: решение уравнений Лотка — Вольтерра

WWW.SCIAM-MAGAZINE.NAROD.RU

ДВИЖЕНИЕ И РАЗМНОЖЕНИЕ РЫБ

Для каждой рыбы в массиве FISH программа строит список соседних свободных позиций и перемещает ее в одну, случайно выбранную из них. Это означает, что в массив FISH заносится -1 в старой позиции и текущий возраст рыбы в ее новой позиции. Массив FISHMOVE изменяется в соответствии с приведенным выше описанием. Если возраст рыбы равен *fbreed*, то программа помещает новую рыбу в старую позицию и приписывает обеим рыбам возраст 0. Новая рыба также регистрируется в FISHMOVE. Если все соседние позиции заняты, то рыба не движется и не размножается.

ОХОТА И РАЗМНОЖЕНИЕ АКУЛ

Для каждой акулы в массиве SHARKS программа строит список соседних позиций, содержащих рыб (если такие позиции есть). Акула случайно выбирает одну из них, перемещается туда и съедает рыбу. Это означает, что программа изменяет массивы SHARKS и SHARKMOVE так же, как это делалось для массивов FISH и FISHMOVE, но не только для них. В рассматриваемую позицию массива FISH следует занести -1, а в ту же позицию массива STARVE — число 0. Если в соседних узлах нет рыб, то акула движется так же, как рыба. Если возраст акулы достигает *sbreed*, то новая акула появляется на свет точно таким же способом, как новая рыба.

ОТОБРАЖЕНИЕ

Программа просматривает оба массива FISH и SHARKS. Она выводит на экран точку для каждой обнаруженной рыбы и нуль для каждой акулы. Отображение можно выполнять полностью, как описано, или же можно разбить это действие на две части, одна из которых будет выполняться после перемещения рыб, другая — после перемещения акул.

Чтобы выполнить первоначальное заселение океана, программисту надо записать цикл, в котором *nfish* раз будет порождаться пара случайных чисел. Эти числа масштабируются по горизонтальному и вертикальному размерам желаемого океана. Таким образом, числа задают случайную позицию, в которую программа помещает рыбу, отмечая ее в массиве FISH и присваивая ей случайный возраст от 0 до *fbreed*. Аналогично расселяются акулы. В обоих случаях проверяется, не занята ли уже выбранная позиция. Присваивание акулам и рыбам случайного возраста ведет к тому, что они потом будут размножаться в случайные моменты времени, что вполне естественно. Без этой меры вы увидели бы, что число акул и рыб вдруг удвоилось бы — крайне удивительная и неестественная картина.

Некоторые начинающие программисты могут найти предшествующее описание слишком общим, чтобы составить по нему ясное представление о том, как же написать программу АКВАТОР. Эти программисты могут начать с составления программы, известной как программа «пьяных блужданий». Такая программа могла бы состоять всего из одного цикла (скажем, цикла *while*), включающего семь инструкций. Эти инструкции записаны ниже на произвольно выбранном алгоритмическом языке. Присваивания обозначаются стрелками, направленными влево; переменные X и Y хранят координаты «шатающегося пьяницы». Они изменяются в соответствии со случайным целым числом, присваиваемым переменной *direction*. В зависимости от того, чему равно это число — 0, 1, 2 или 3, «пьяница», точка на экране сдвинется на «север», «восток», «юг» или «запад».

```
direction ← целая часть
от (random × 4)
if direction = 0 then X ← X + 1
if direction = 1 then X ← X − 1
if direction = 2 then Y ← Y + 1
if direction = 3 then Y ← Y − 1
отобразить (X, Y)
```

Предположим, что генератор случайных чисел порождает десятичное число *random* между 0 и 1, тогда данный алгоритм преобразует его в число в диапазоне от 0 до 3.999. Целая часть полученного числа должна равняться 0, 1, 2 или 3.

Я не берусь утверждать, что пятно света, «разгуливающее» по экрану, соответствует экологической «драме» акул и рыб, однако написание этой программы позволит лучше уяснить, как построить отдельные части программы АКВАТОР.

Опытные программисты, могут поразмыслить над другими путями реализации программы АКВАТОР. Можно значительно уменьшить число выполняемых действий, если для хранения информации об акулах и рыбах применить связанные списки. В этом случае время, требуемое для одного цикла вычислений, будет пропорционально числу имеющихся акул и рыб, а не размеру всего океана.

Программа АКВАТОР может рассказать кое-что и о наших, земных животных. Мы знаем, что небольшие популяции сталкиваются с реальной угрозой вымирания и, даже если ни хищники, ни жертвы не исчезают полностью, их численность почти наверняка периодически колеблется. В простой экосистеме численности хищников и жертв иногда испытывают циклические колебания. Так ведут себя популяции зайца и рыси в канадском приполярье, численности которых регистрировались с 1847 по 1903 г. специалистами из Hudson's Bay Company (см. рисунок на с. 82). Числа представляют количество особей каждого вида, добытых за каждый год. Эти числа, по-видимому, пропорциональны действительной численности видов в рассматриваемый период. Если это так, то циклы легко объясняются: рыси поедают все еще возрастающую популяцию зайцев, потом численность зайцев начинает уменьшаться по мере роста числа рысей. Вскоре для рысей становится меньше пищи, и они начинают погибать от голода или приносить меньший приплод или же и то и другое вместе. Когда число рысей падает, вновь начинают размножаться зайцы.

Эти графики можно сопоставить с гладкими кривыми, изображающими решение уравнений Лотка—Вольтерра. Впервые эти уравнения были получены в 1931 г. В. Вольтерра, итальянским математиком*. Они основаны на предположении, которое можно было бы охарактеризовать как «непрерывную охоту непрерывного хищника за непрерывной жертвой». Решения уравнений демонстрируют циклические изменения, что на первый взгляд отвечает эмпирическим данным по рысям и зайцам. Среди биологов, однако, нет единогласия в вопросе о том, допустимо ли объяснять численности рысей и зайцев на основе таких простых соображений. Прежде всего в охоте на зайцев участвуют еще по крайней мере два вида «хищников» — микробы и люди.

Как бы то ни было, имело смысл собрать статистические данные по акулам и рыбам на Аква-Торе, и мы с Магги так и поступили. Недавно полученные нами графики численности акул и рыб представляются более похожими на графики для рысей и зайцев, чем решения уравнений Лотка—Вольтерра. Мы до сих пор думаем над долговременной нестабильностью, характерной для некоторых комбинаций параметров. Может быть, кто-нибудь из читателей, работающих над собственной версией программы АКВАТОР, сумеет продвинуться дальше в этом вопросе. Существует ли какое-либо общее правило, позволяющее для заданного набора параметров предсказать стабильность или нестабильность получившейся экосистемы? В какой степени циклические колебания описываются уравнениями Лотка—Вольтерра?

Причина тороидальности океана Аква-Тора очень проста: составление программы сильно облегчается, если у

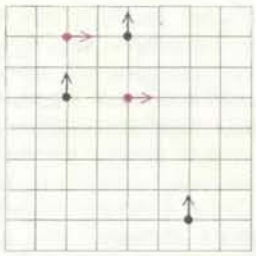
---

* См.: Вольтерра В. Математическая теория борьбы за существование. — М.: Наука, 1976. — *Прим. перев.*

WWW.SCIAM-MAGAZINE.NARODRU

океана нет границ или берегов. Если океан имеет в ширину, скажем, 32 единицы, то в качестве X-координат акул и рыб можно просто использовать числа по модулю 32. Если кто-то из них в течение одного хронона имеет X-координату 31 и наблюдается справа на экране, то в следующем хрононе X-координата вполне может стать равной $32=0$ и акула или рыба появится слева. Такая же система действует по вертикали.

Тороидальный океан Аква-Тора является источником некоторых весьма странных эффектов, примеры которых содержатся в следующих задачах. Первый из этих эффектов возник вследствие ошибки в ранней версии моей программы АКВАТОР. Эта ошибка привела к тому, что все рыбы в течение хронона проплывали одну единицу на «север», а все акулы — одну единицу на «восток». Таким образом, акула получала возможность съесть рыбу, только если они обе попадали в один узел. Сколько рыб в изображенном ниже океане никогда не будут съедены?



В следующей задаче акулы и рыбы наделяются «интеллектом». Допустим, что каждая акула и рыба при переходе в соседний узел «оглядывается по сторонам». Тогда оказывается, что одна рыба, если она достаточно «умна», всегда может уйти от одной акулы независимо от «умственных способностей» последней. Если в тороидальном океане Аква-Тора за одной-единственной рыбой охотятся две акулы, то исход может быть двоякий. Пусть акулы и рыбы будут сколь угодно «умны», причем акулам даже будет разрешено координировать свои действия. Сможете ли вы в этих условиях найти выход для рыбы? Ответ не зависит от размеров океана.

СТАТЬЯ о персептронах («Computer Recreations», September)* привела некоторых читателей к мысли об их использовании, других же побудила заняться собственными исследованиями. Эд Маннинг из Стратфорда (шт. Коннектикут) построил «персептрон сортов» (perceptron of sorts), придуманный 10 лет назад для того, чтобы преобразовывать реальные образы в дискретные квадратики на сетчатке персептрона. Маннинг был одним из многих, кто заметил ошибку в окошечном персептроне для распознавания нескольких прямоугольников, изображенном в сентябрьском номере журнала: четыре последних элемента в списке подобразов демона должны быть наполовину голубые и наполовину белые. Маннинг интересуется, не была ли эта ошибка специально задумана как испытание для читателей. Не могу удержаться от соблазна и отвечаю, что так оно и было.

Дж. Стормо, научный сотрудник факультета молекулярной, клеточной и эволюционной биологии Колорадского университета в Боулдере, использовал идею персептрона в автомзтическом распознавании образов. Он построил весовую функцию персептрона для распознавания участков связывания в последовательности нуклеотидов транспортных РНК. Чтобы его персептрон работал эффективно, Стормо воспользовался теоремой сходимости. Результаты оказались весьма обнадеживающими: персептрон распознавал участки связывания весьма успешно.

Любой окошечный персептрон, список подобразов которого включает сплошь белое или сплошь черное окошко, является «хорошим» персептроном. Это отмечает Л. Хилл из Вениса (шт. Калифорния). Его также интересует вопрос, является ли допустимым, чтобы сетчатка была сплошь белая или сплошь черная. Ответ зависит от распознаваемых образов. В случае персептрона для распознавания нескольких прямоугольников представляется разумным считать сплошь черную сетчатку просто одним большим прямоугольником.

Наибольших успехов в предложенном мной мини-исследовании добился К. Руссо из Вычислительного центра колледжа Линчберга (шт. Виргиния). Он решил исключить окошечные персептроны со сплошь белыми или сплошь черными подобразами. Ему удалось охарактеризовать «хорошие» персептроны (т.е. такие, которые распознают хотя бы один образ). Для этого используются трансляционные соотношения между подобразами в списке персептрона. Если сдвинуть окошечный подобраз на одну клетку в любом из четырех основных направлений, то снова должен получиться какой-то подобраз из списка. Затем Руссо рассмотрел минимальные окошечные персептроны, список подобразов которых не может быть сокращен без ущерба их распознавательной способности. Из таких персептронов, как из кирпичиков, строятся все «хорошие» персептроны. Руссо написал программу для ЭВМ, которая нашла все минимальные окошечные персептроны с размером списка от 2-го до 5-го порядка. Нет ни одного минимального персептрона 6-го порядка. Руссо бросил всем вызов, обратив поставленную мной задачу: я предлагал читателям найти образ, распознаваемый данным персептроном. Руссо же предлагает отыскивать персептрон, распознающий данный образ.

Д. Эванс из Хартфорда (шт. Коннектикут) возлагает вину за неудачи персептронов на ограничение, присущее двухуровневой иерархии локальных демонов, отчитывающихся перед одним главным демоном. Введя некое демоническое промежуточное управление, Эванс преодолел открытую Минским и Пэйпертом «слабость» обычных персептронов, которые неспособны распознавать связность. Демоны нижнего уровня образуют своего рода сетчатку, в которой черное или белое состояние клетки отвечает наличию или отсутствию сигнала от определенного демона. Демоны второго уровня следят за образами, созданными демонами нижнего уровня, и посылают сигнал о наличии или отсутствии подобраза главному демону. Трехуровневое устройство способно распознать, какой из четырех пробных образов является связным, а какой нет.

* См.: «В мире науки», 1984, №11, с. 127. — *Прим. ред.*

Издательство **МИР** *предлагает:*

М. Лонгейр
АСТРОФИЗИКА ВЫСОКИХ ЭНЕРГИЙ
Перевод с английского

Видный английский астрофизик в своем курсе лекций, прочитанных в Кембриджском университете, подробно рассматривает процессы во Вселенной, происходящие с участием частиц и фотонов высокой и сверхвысокой энергии. Автор описывает протонную и электронную компоненты космических лучей, космическое рентгеновское и гамма-излучение, а также такие экзотические объекты, как пульсары, квазары, активные ядра галактик — наглядные свидетельства взрывных процессов чудовищной интенсивности.

Для астрономов и физиков — теоретиков и экспериментаторов, как специалистов, так и интересующихся новой проблематикой студентов.

1984, 26 л. Цена 4 р. 20 к.

ИЗДАТЕЛЬСТВО «МИР»